AI TOOLS WIKI · MÉMO

# Agent Browser *mémo*

Les commandes à connaître pour **Agent Browser**, le CLI de Vercel Labs qui permet à Claude Code ou Codex de piloter un vrai navigateur Chrome : ouvrir une page, lire sa structure, cliquer, remplir, capturer.

**Famille** Navigateur et tests · **Éditeur** Vercel Labs · **Licence** Apache-2.0
**Note du wiki** 11/15 · **Commandes vérifiées le** 25 septembre 2026
**Fiche, notes et avis** aitoolswiki.viellard.fr/outils/navigateur-et-tests/agent-browser

## Installation

| | |
|---|---|
| `npm install -g agent-browser` | ou : brew install agent-browser / cargo install agent-browser |
| `agent-browser install` | télécharge Chrome for Testing (première fois) |
| `npx skills add vercel-labs/agent-browser` | Ajouter la skill à Claude Code, Codex, Cursor… |

La skill installée est un simple pointeur vers `agent-browser skills get core` : les instructions suivent toujours la version du CLI installée. Ne pas copier le `SKILL.md` depuis `node_modules`, il deviendrait obsolète.

## Le workflow de base (à mettre dans CLAUDE.md)

1. `agent-browser open <url>` – ouvrir la page
2. `agent-browser snapshot -i` – lister les éléments interactifs, chacun avec une référence (`@e1`, `@e2`…)
3. `agent-browser click @e1` / `agent-browser fill @e2 "texte"` – agir via les références
4. **Refaire un snapshot après chaque changement de page**

## Commandes principales

### Naviguer et lire

| | |
|---|---|
| `agent-browser open <url>` | Lance le navigateur et ouvre l'URL. |
| `agent-browser snapshot -i` | Arbre d'accessibilité limité aux éléments interactifs, avec références. **La commande la plus utile pour un agent.** |
| `agent-browser snapshot -i -c -d 5` | Idem, compacté (`-c`) et limité en profondeur (`-d`) pour économiser des tokens. |
| `agent-browser read [url]` | Récupère le texte lisible d'une page (Markdown si disponible), sans lancer Chrome. |
| `agent-browser get text @e1` | Texte d'un élément. Aussi `get url`, `get title`, `get value`, `get html`. |

## Commandes principales (suite)

| | |
|---|---|
| `agent-browser back` / `forward` / `reload` | Navigation. |

### Agir

| | |
|---|---|
| `agent-browser click @e2` | Clique sur un élément. |
| `agent-browser fill @e3 "a@b.com"` | Vide puis remplit un champ. |
| `agent-browser type @e3 "texte"` | Tape dans un champ sans le vider. |
| `agent-browser press Enter` | Appuie sur une touche. |
| `agent-browser select @e4 "valeur"` | Choisit une option de liste déroulante. |
| `agent-browser upload @e5 fichier.pdf` | Envoie un fichier. |
| `agent-browser find role button click --name "Submit"` | Trouve un élément par son rôle, son texte ou son label, puis agit. Plus robuste qu'un sélecteur CSS. |
| `agent-browser wait --text "Bienvenue"` | Attend un texte, un élément (`wait "#id"`), une URL (`--url`) ou un état de chargement (`--load`). |

### Capturer et déboguer

| | |
|---|---|
| `agent-browser screenshot page.png` | Capture d'écran (`--full` pour la page entière). |
| `agent-browser screenshot --if-changed` | Ne renvoie l'image que si elle a changé. **Recommandé pour économiser des tokens.** |
| `agent-browser screenshot --annotate` | Capture avec les éléments numérotés. |
| `agent-browser console` / `errors` | Messages console et erreurs JavaScript. |
| `agent-browser network requests --filter api` | Requêtes réseau suivies, filtrables. |
| `agent-browser diff snapshot` | Compare avec le snapshot précédent. |
| `agent-browser a11y` | Audit d'accessibilité (axe-core embarqué). |
| `agent-browser vitals` | Web Vitals (LCP, CLS, INP…). |

### Enchaîner et gérer

| | |
|---|---|
| `agent-browser batch "open https://…" "snapshot -i" "screenshot"` | Plusieurs commandes en un seul appel : moins de démarrages de process. |
| `agent-browser --session agent1 open …` | Sessions isolées (cookies, onglets, historique séparés). |
| `agent-browser --profile Default open …` | Réutilise les connexions d'un profil Chrome existant. |

## Commandes principales (suite)

| | |
|---|---|
| `agent-browser state save auth.json` / `state load auth.json` | Sauvegarde et recharge un état connecté. |
| `agent-browser close` | Ferme le navigateur (`--all` pour toutes les sessions). |
| `agent-browser doctor` | Diagnostique l'installation. |
| `agent-browser mcp` | Démarre le serveur MCP (profil d'outils `core` par défaut). |

## Sécurité

| Option | Effet |
|---|---|
| `--allowed-domains "example.com,*.example.com"` | Limite la navigation à ces domaines. |
| `--content-boundaries` | Encadre le contenu des pages pour que le modèle le distingue des instructions. |
| `--max-output 50000` | Plafonne la taille des réponses. |
| `--confirm-actions eval,download` | Demande confirmation avant les actions sensibles. |
| `agent-browser auth save <nom>` / `auth login <nom>` | Coffre-fort d'identifiants chiffré : le modèle ne voit jamais le mot de passe. |

## Conseils d'utilisation

- **Utiliser les références (`@e1`) plutôt que des sélecteurs CSS** : elles viennent du snapshot et sont plus fiables.
- **Toujours refaire un snapshot après une navigation** : les références d'un ancien snapshot ne sont plus valides.
- **Économiser le contexte** : `snapshot -i -c`, `screenshot --if-changed` et `batch` réduisent nettement les tokens consommés.
- **Clic qui échoue** : souvent une bannière de cookies ou une modale devant l'élément. La fermer, puis refaire un snapshot.
- **Attendre un état précis** (texte, URL, élément) plutôt que `--load networkidle`, qui peut ne jamais se terminer sur les pages temps réel.
- **Pour les sites sensibles**, combiner `--allowed-domains` et le coffre-fort `auth` : les pages web peuvent contenir des injections de prompt.

## Sources (documentation officielle)

- README agent-browser
- Site agent-browser.dev
- Browser Automation in Claude Code: 5 Tools Compared